

Trabajo Integrador 2023

Seminario de Lenguajes Opción Python

El trabajo integrador para este año es desarrollar una aplicación que permite crear imágenes a partir de otras combinadas con posibilidad de agregar texto. Esta aplicación, a la que denominaremos **UNLPImage**, permitirá:

- crear un collage con fotos o imágenes disponibles en nuestra computadora,
- generar un meme a través de combinación de imágenes con texto y emojis.

El propósito de este trabajo es que los estudiantes puedan aplicar los conocimientos adquiridos a lo largo del curso en el desarrollo de un proyecto completo y funcional.

UNLPImage será una aplicación de escritorio que además de generar imágenes permitirá clasificar imágenes y acceder a las imágenes almacenadas en nuestra computadora a través de información sobre las mismas.

La interfaz gráfica de usuario deberá ser amigable y fácil de usar para permitir la interacción de manera intuitiva. Se espera que los estudiantes demuestren su capacidad para aplicar sus conocimientos en Python de manera creativa y práctica para resolver problemas concretos.

El desarrollo se organiza en 2 etapas que se entregarán de acuerdo al cronograma ya publicado.

En este documento describiremos los distintos aspectos que debe cubrir el trabajo a realizar y hacia el final se detallan las consideraciones y requerimientos de esta entrega.

Para el desarrollo de la aplicación se deberá:

- Aplicar los conocimientos aprendidos en la teoría sobre el lenguaje, poniendo especial atención en tratar de desarrollar código de calidad teniendo en cuenta las guías de estilo de Python.
- Definir una estructura de carpetas que permita estructurar el código de forma prolija.
- Desarrollar la aplicación con un equipo de trabajo donde deberán dividir las tareas y ayudarse mutuamente para completar lo especificado.
- Utilizar **PySimpleGUI** para las interfaces gráficas.
- Generar los archivos CSV que se requiera.
- Documentar el código usando docstrings en las funciones y clases.
- Incluir un archivo LICENSE con los nombres de los autores y la licencia.

UNLPImage

El archivo principal de la aplicación deberá llamarse **unlpimage.py** la cual contará con las siguientes pantallas que describiremos más adelante:

- A. Inicio
- B. Nuevo perfil
- C. Menú principal
- D. Editar perfil
- E. Configuración
- F. Generador de memes (sólo interfaz)
- G. Generador de collage (sólo interfaz)
- H. Etiquetar imágenes

Las pantallas que se indican (sólo interfaz) deberán desarrollar sólo los componentes sin la funcionalidad correspondiente, se deberá poder ingresar a la pantalla y salir.

Inicio

Al ejecutar la aplicación UNLPImage, la primera pantalla (**A - Inicio**), solicitará que se seleccione su perfil de una lista ya o que registre uno nuevo en caso de no existir.

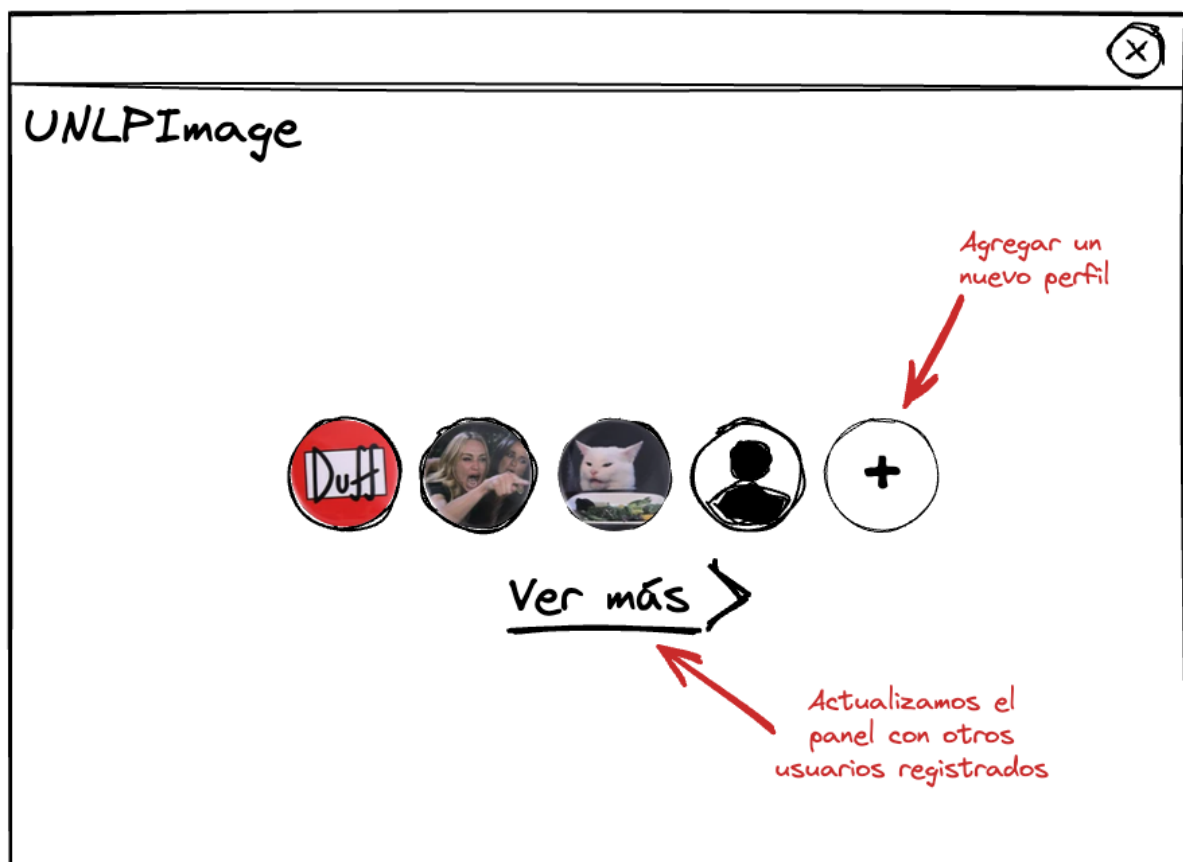


Fig. 1. Ejemplo de pantalla de inicio.

Como vemos en la imagen (Fig. 1), dentro de esta primer pantalla se mostrará:

- una lista con los avatares de los perfiles registrados,
- un botón para agregar un nuevo perfil,
- opcionalmente un botón que permita mostrar más perfiles (en caso que haya registrados) para elegir.

Una vez que se selecciona el perfil en la aplicación se accede al menú principal de la aplicación (**C - Menú principal**) donde se pueden visualizar las opciones de la aplicación.

Para el caso en que se quiera agregar un nuevo perfil por primera vez se mostrará la pantalla de carga de los mismos (**B - Nuevo perfil**).

Nuevo perfil

En esta pantalla se deberá visualizar un formulario que permita la carga de los datos necesarios para dar de alta un perfil.

Nuevo perfil

< Volver

Nick o alias

Nombre

Edad

Genero autopercebido

Seleccione una opción

Otro

Complete el género

Seleccionar avatar

Guardar

Fig. 2. Ejemplo de pantalla B - Nuevo perfil.

Como se muestra en la imagen (Fig. 2), se solicitarán los siguientes campos:

- nick o alias (deberá ser único)
- nombre
- género autopercebido
- edad
- avatar (será una imagen a elección del usuario)

Todos los campos son obligatorios para realizar el alta de un nuevo perfil. En el caso del avatar, si el usuario no selecciona una imagen el sistema debe mostrar una imagen por defecto.

Estos datos son importantes dado que nos permitirán luego realizar estadísticas del uso de la aplicación posteriormente. Luego de agregar el nuevo perfil se seleccionará automáticamente y se debe pasar al menú (**C - Menú principal**).

El sistema deberá almacenar los perfiles en un archivo JSON. En este archivo se deberán agregar los nuevos perfiles que se registren.

Menú Principal

En el menú principal se deberán mostrar todas las opciones que permite realizar nuestra aplicación.

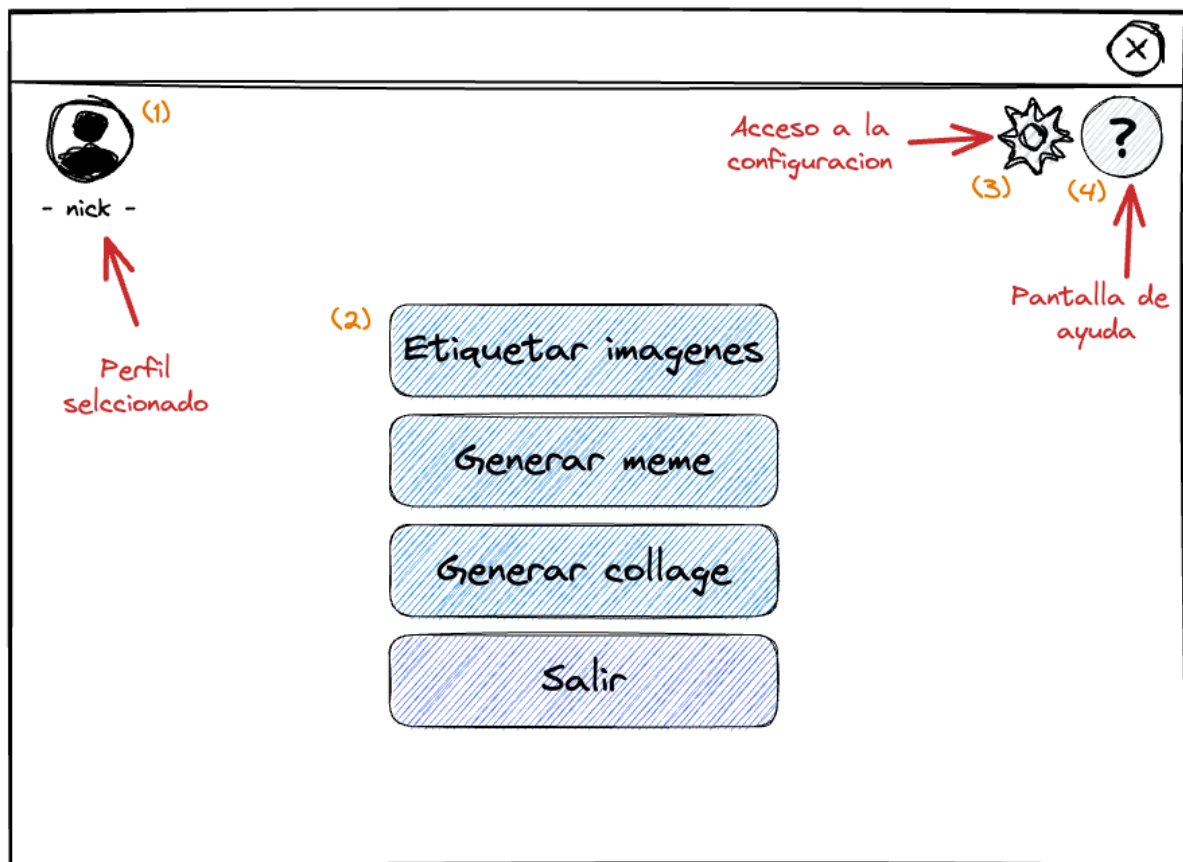


Fig. 3. Ejemplo de pantalla C - Menú principal.

Esta pantalla debe permitir una navegación fluida entre las diferentes funcionalidades de la aplicación. Como se puede observar en la imagen (Fig. 3), dentro de este menú se deberán encontrar las siguientes opciones:

1. **Información de perfil seleccionado:** se mostrará el avatar y el nombre del perfil que se seleccionó. Y al hacer clic en la imagen, desplegará la ventana para editar el perfil (**D - Editar perfil**).
2. **Menú de opciones:** es el menú que muestra permite el acceso a las funcionalidades principales de la aplicación.
3. **Configuración:** botón que permite acceder a la ventana de configuración de la aplicación.
4. **Ayuda:** simplemente será una pantalla emergente con una breve explicación de la funcionalidad provista por la aplicación.

Editar Perfil

Esta pantalla despliega un formulario muy similar a la ventana de creación de un nuevo perfil (ejemplo en Fig. 2). En este caso el formulario deberá contener todos los datos del usuario seleccionado y le permitirá la edición.

El único atributo de los solicitados al perfil que no se puede editar será el nick o alias.

Configuración

Esta opción abrirá una pantalla nueva (**E - Configuración**) donde se podrá configurar las siguientes opciones de la aplicación:

- **Directorio de nuestro repositorio de imágenes:** en este directorio es donde se encuentran todas las imágenes que podemos utilizar en nuestra aplicación.
- **Directorio de almacenamiento de collages:** donde se almacenarán los collages generados.
- **Directorio de almacenamiento de memes:** donde se almacenarán los memes generados.

Las configuraciones se aplican a nivel de aplicación, es decir todos los perfiles que ingresen comparten los mismos datos y cualquiera podrá editar estas opciones.

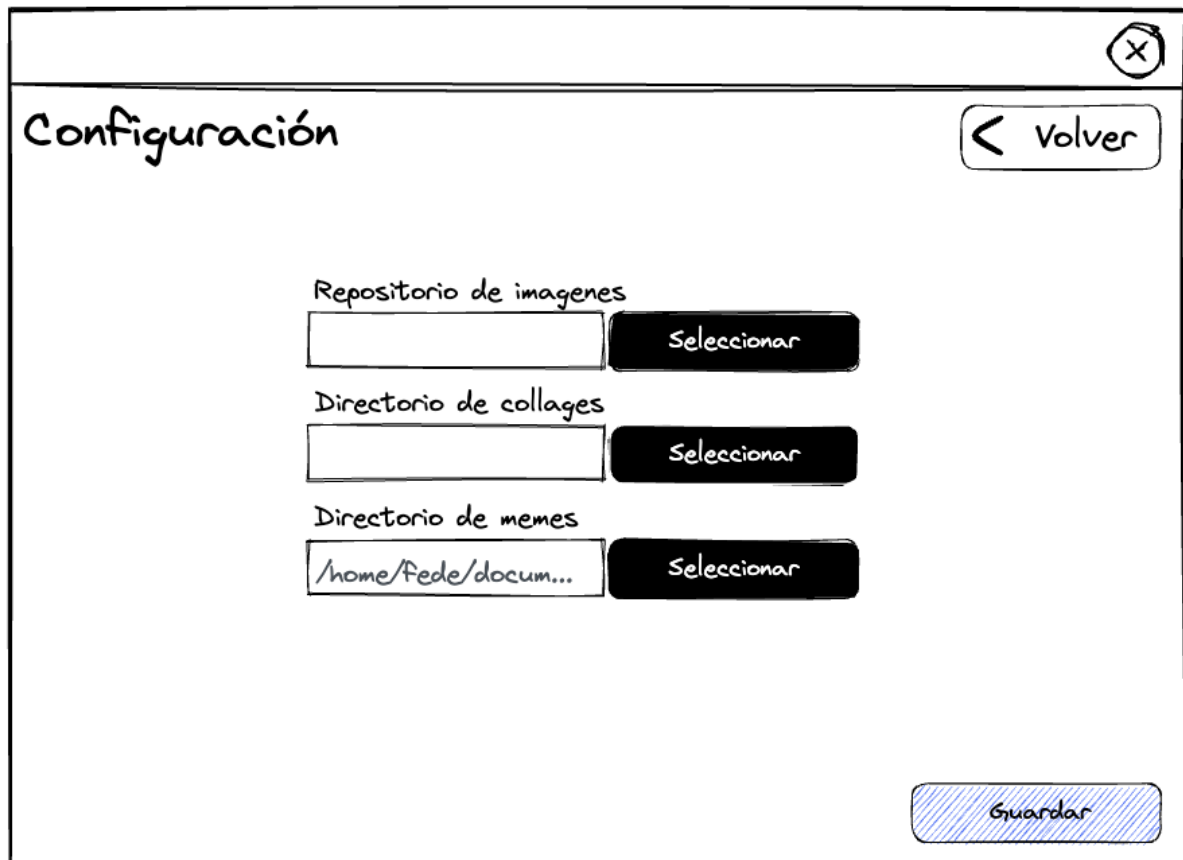


Fig. 4. Ejemplo de pantalla E - Configuración.

Generador de memes (sólo interfaz)

Esta herramienta fundamentalmente realiza el proceso de generación a partir de varios componentes:

- una imagen base del meme,
- una fuente para introducir el texto.

Para esta funcionalidad se tomarán del **directorio de repositorio**, las imágenes base (sin texto) que se utilizarán para generar los memes. A las mismas se les agrega el texto en los diferentes sectores de la imagen. De todas formas **para esta entrega sólo será necesaria la interfaz gráfica** que permita ingresar a la pantalla y salir.

Generador de collage (sólo interfaz)

Para la generación de collage se debe disponer de varias plantillas que darán la especificación de la ubicación de las imágenes para generar el collage.

Al igual que la pantalla de generación de memes ésta no tendrá ninguna funcionalidad implementada. La idea es que realicen una interfaz gráfica para entrar y salir de la ventana.

Etiquetar imágenes

Esta funcionalidad permite clasificar las imágenes contenidas en el directorio que se configuró como repositorio de imágenes para la aplicación.

La aplicación deberá permitir agregar tags o etiquetas e identificar los metadatos de las imágenes en nuestra carpeta repositorio. Estas características de la imagen facilitarán la búsqueda y la recuperación tanto en la generación de memes (**F - Generador de memes**) como para la generación de collages (**G - Generador de Collages**).

Esta información deberá almacenarse **en un archivo CSV** que hará de base de datos todas las imágenes clasificadas.

La Fig. 5 muestra una posible visualización, donde se puede seleccionar una imagen (en este caso en un árbol de archivos y directorios) y para la misma permite la visualización de las etiquetas (si las tiene) y metadatos, con la posibilidad de agregar nuevas etiquetas y otras características a la misma.

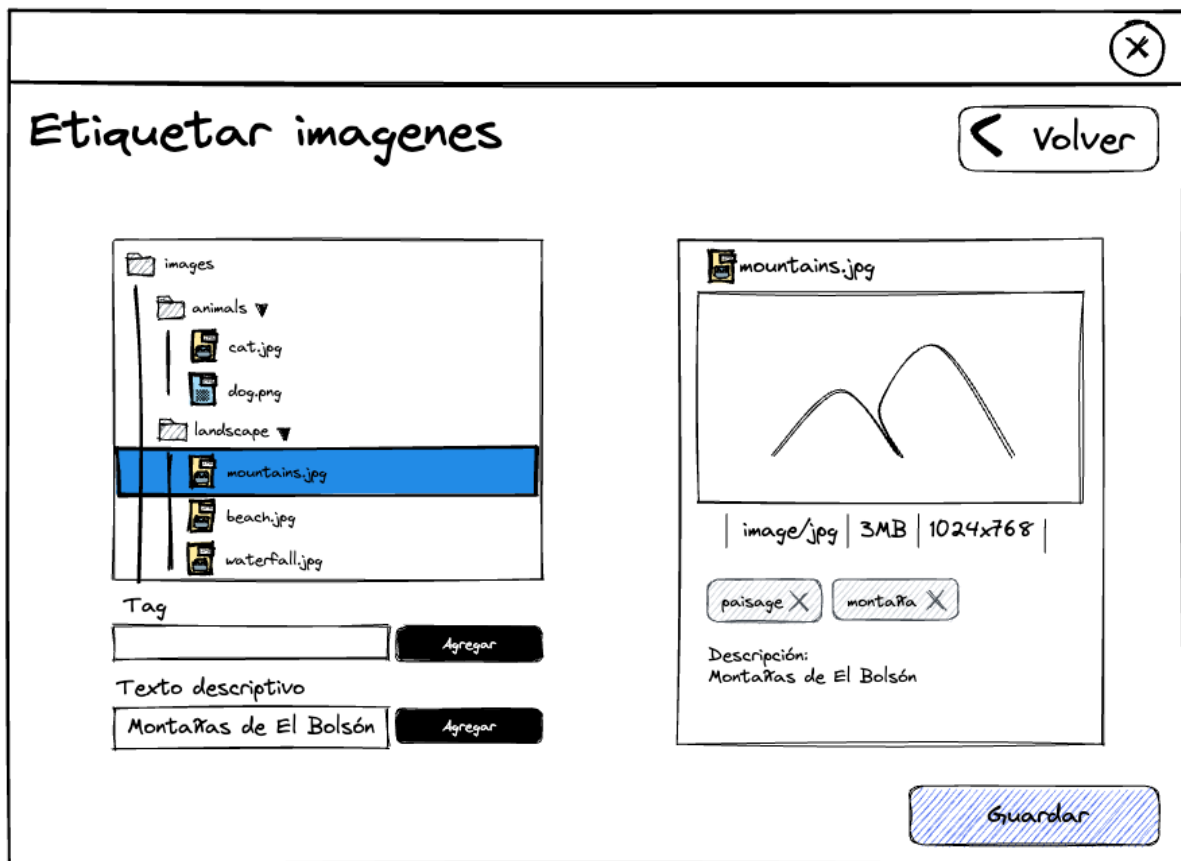


Fig. 5. Ejemplo de pantalla H - Etiquetar imágenes.

Como dijimos, esta herramienta también deberá permitir a los usuarios identificar las características de las imágenes. Esto significa que quien esté realizando el proceso de clasificación puede obtener información detallada sobre cada imagen, como la fecha en que fue tomada, la ubicación, la cámara utilizada, mimetype, resolución, etc. de los metadatos.

Esta información es muy útil para poder realizar una mejor clasificación de las imágenes y facilitar su accesibilidad en el futuro.

Además nos permitirá crear un dataset con las distintas características que tiene nuestro repositorio de imágenes para poder hacer análisis de datos sobre ellos.

La información que obligatoriamente deben guardarse son:

- ruta relativa a la imagen*
- texto descriptivo
- resolución* (alto y ancho)
- tamaño*
- tipo* (mimetype)
- lista de tags
- último perfil que actualizó
- fecha de última actualización

Las características marcadas con * deberán ser extraídas directamente de los metadatos de los archivos y las otras deberán ser completadas por el usuario (texto descriptivo y lista de tags). Por último, los campos “último perfil que actualizó” y “fecha de última actualización” deberán completarse de forma automática por medio del sistema al momento de realizar la acción.

Al abrir la pantalla de edición de características y seleccionar una imagen dentro de nuestro repositorio que se quiera seleccionar, los usuarios verán un formulario con los metadatos asociados a la imagen seleccionada, incluyendo información como la fecha de creación, el tamaño del archivo, el formato y otros detalles relevantes que considere. Desde aquí, los usuarios podrán editar los valores de las características existentes si la imagen fué clasificada anteriormente o agregar la nueva información si es la primera vez que se clasifica la imagen.

Además, la herramienta debe permitir a los usuarios agregar o quitar tags de la imagen seleccionada. Los usuarios pueden agregar nuevas etiquetas para organizar mejor sus imágenes, o eliminar etiquetas existentes que ya no sean relevantes.

Una vez que los usuarios hayan realizado todos los cambios necesarios para la imagen seleccionada, podrán guardar los cambios para que se reflejen de forma permanente en el archivo CSV. Esta funcionalidad proporciona una solución práctica para mantener una biblioteca de imágenes organizada y fácilmente accesible.

Para poder evaluar esta funcionalidad le solicitamos que suban las imágenes que se encuentran en el archivo CSV al repositorio. Para evitar sobreuso del almacenamiento remoto no exceder los 10MB.

Logs del sistema

La aplicación además deberá mantener un registro de las principales interacciones que se realicen dentro de la misma. Esto nos generará un registro del uso de la misma y será información muy útil para realizar un posterior análisis.

Los eventos que deberá registrarse para esta primera etapa serán las siguientes operaciones:

- cambio en la configuración del sistema,
- nueva imagen clasificada,
- modificación de imagen previamente clasificada.

Estos logs deberán almacenarse en un archivo CSV. Donde cada línea del archivo tendrá como mínimo los siguientes campos:

- fecha y hora (timestamp),
- nick o alias del perfil que realizó la acción,
- operación (las 3 mencionadas anteriormente).

Consideraciones de la entrega y defensa

Si bien el trabajo es grupal, **la defensa es INDIVIDUAL**.

La defensa se realiza dentro del horario de consulta inmediatamente posterior a la fecha de entrega. La misma consiste en tener un **encuentro con el ayudante asignado de forma virtual** donde el mismo va a realizar una serie de preguntas sobre la entrega. Estas preguntas serán destinadas a los distintos miembros del grupo. La idea es que cada estudiante responda las preguntas destinadas a su persona teniendo espacio individual para responder.

El desempeño en las respuestas, sumado a la participación en las consultas y el registro commits que se puede observar en Gitlab serán los parámetros que el ayudante tendrá en cuenta a la hora de definir la nota de la entrega para cada integrante.

Esto quiere decir que por más que sea una entrega grupal, las notas entre los miembros de los grupos pueden ser diferentes.

Repetimos que es importante también tener **participación activa en las consultas prácticas y en el repositorio de GitLab** para que el ayudante tenga un panorama conceptual completo del conocimiento de cada participante del grupo.

Al trabajar con imágenes el tamaño del repositorio puede crecer considerablemente, por lo tanto le pedimos no exceder los 10MB.

Fechas de entrega

La fecha de entrega tiene fecha límite el **Viernes 12 de Mayo a las 23:59hs y la defensa individual se desarrollará la semana siguiente en el horario de la práctica correspondiente (Jueves 18 o Viernes 19 según corresponda).**

Dentro del portal de Cátedras encontrarán 2 tareas referidas a esta entrega.

- En una tendrán la devolución del trabajo grupal. El ayudante les dará la devolución y sugerencias sobre el código desarrollado.
- La otra será para que cada participante tenga el feedback de la evaluación individual de la defensa.

Dado que el código se completa dentro del repositorio de Gitlab no hará falta que realicen ninguna entrega en estas tareas.

Criterios de evaluación

- Funcionalidad implementada de acuerdo al enunciado.
- Cumplimiento de las consideraciones planteadas.
- Código subido al repositorio de GitLab indicado.
- Participación durante el desarrollo del trabajo. Esto incluye tanto la asistencia a las consultas (virtual o presencial) y las contribuciones en el repositorio de GitLab.

- Desempeño en la defensa donde deberán demostrar los conocimientos utilizados para la realización del trabajo.

Puntos de la primera entrega (en la defensa individual)

- **Puntos de cursada:** 35 puntos.

Recomendaciones sobre el trabajo en grupo

- **Establecer un plan de trabajo:** es importante que los estudiantes establezcan un plan de trabajo detallado y que definan claramente las tareas que cada uno va a realizar. Esto les permitirá trabajar de manera más eficiente y organizada.
- **Releer el enunciado regularmente** para recordar las especificaciones y evitar confusiones.
- **Utilizar herramientas de comunicación:** los estudiantes pueden utilizar herramientas de comunicación como Discord para mantenerse en contacto y coordinar el trabajo. Es importante que todos estén en la misma página y se comuniquen con regularidad.
- **Utilizar un flujo de trabajo en Git:** los estudiantes deben familiarizarse con un flujo de trabajo en Git. Es importante que los estudiantes se comuniquen y se pongan de acuerdo en el flujo de trabajo que van a utilizar.
- **Establecer reuniones regulares:** los estudiantes pueden establecer reuniones regulares para discutir el progreso del proyecto y resolver cualquier problema que haya surgido. Esto les permitirá mantenerse al día y abordar los problemas de manera oportuna.
- **Utilizar comentarios en el código:** es importante que los estudiantes utilicen comentarios en el código para explicar qué hace cada sección y cómo funciona el programa. Esto facilitará la comprensión del código para los demás integrantes del grupo.
- **Fomentar el trabajo en equipo:** los estudiantes deben fomentar el trabajo en equipo y ayudarse mutuamente. Es importante que estén dispuestos a dar y recibir retroalimentación constructiva.